

# Ein Modell zur Simulation der Bewegung von Schwärmen

Ausarbeitung im Rahmen des Proseminars 2004

Christian Tomm

Es wird ein Modell vorgestellt, mit dem eine natürliche Animation von Tierschwärmen ermöglicht wird. Das Modell basiert auf der Kombination von mehreren einfachen, grundlegenden Verhalten. Diese Verhaltensweisen lassen sich von der räumlichen Dimension unabhängig und können zur Modellierung von allen Arten von Schwärmen eingesetzt werden.

## Inhaltsverzeichnis

1. Einleitung. . . . .	2
2. Grundlagen. . . . .	2
3. Grundlegende Verhaltensweisen der Bewegung. . . . .	3
<b>3.1 Ein Fahrzeugmodell.</b> . . . . .	3
<b>3.2 Verschiedene Verhaltensweisen.</b> . . . . .	4
4. Schwarmverhalten. . . . .	10
<b>4.1 Einzelbedürfnisse.</b> . . . . .	10
<b>4.2 Kombinieren von Bedürfnissen.</b> . . . . .	10
<b>4.3 Wahrnehmung.</b> . . . . .	11
<b>4.4 Ausweichen.</b> . . . . .	11
<b>4.5 Gesteuerte Schwärme</b> . . . . .	12
5. Spezielle Anpassung des Modells. . . . .	12
6. Zusammenfassung. . . . .	12
7. Literaturliste. . . . .	13

## 1. Einleitung

Herden und Schwärme sind überaus interessante Gebilde. Obwohl ein Schwarm aus vielen einzelnen Tieren mit unabhängigen Bewegungen besteht ist der Gesamteindruck eines Schwarms doch sehr gleichförmig und elegant. Kein Tier hat einen festen Platz und doch ist der Schwarm ein zusammenhängendes Objekt ohne extreme Ausstülpungen oder Lücken. Diese Komplexität, die doch auf einfachen Prinzipien beruht, ist einer der interessantesten Punkte an der Bewegung von Schwärmen.

Die Prinzipien die in dem vorgestellten Modell angewendet werden sind allerdings nicht auf eine bestimmte Art von Schwärmen beschränkt. Stattdessen können mit gewissen Modifikationen alle Möglichen Arten von Schwärmen, Herden oder ähnlichem simuliert werden.

Der Einfachheit halber wird hier nur von Schwärmen gesprochen, gemeint sind damit aber alle Arten von Herden, Schwärmen oder ähnlichen Gruppierungen.

Ein wichtiger Anwendungsbereich für ein solches Modell ist zum Beispiel die Animationstechnik. Hier kann man entweder jeden Schritt eines Vogelschwarms festlegen oder aber dem Computer eine natürliche Bewegung simulieren lassen. Die Simulation ist aus verschiedenen Gründen effizienter. Man kann z.B. bei einem statischen Modell zwar jeden Vogel mit einem eigenen Weg beschreiben, aber muss dann bei einer Änderung im Szenario jeden Vogel einzeln anpassen. Noch deutlicher ist dieses Problem bei Computerspielen, wo dieses Modell auch Anwendung findet. Hier wird das Szenario sogar variabel erzeugt und von daher können die Pfade gar nicht fest beschrieben werden.

Craig W. Reynolds hat 1987 dieses Modell entwickelt. Es wurde erstmals in dem Film Batman 2 verwendet, um Fledermausschwärme und Horden von Pinguinen zu erstellen. Auch die Stampede in „Der König der Löwen“ wurde mit Hilfe dieses Modells animiert.

Die Grundlagen für diese Ausarbeitung bilden zwei Arbeiten von Craig W. Reynolds ([1], [2]).

Im Allgemeinen basiert dieses Modell darauf, dass man ein Schwarmmitglied mit einigen Verhaltensweisen modelliert und dann mehrere dieser Schwarmmitglieder erzeugt, die dann in durch ihre Reaktionen aufeinander ein Schwarmverhalten zeigen.

In dieser Ausarbeitung werden zunächst verschiedene Bewegungsverhalten erklärt, die für autonome Bewegungen unerlässlich sind, dann wird näher auf die Mechanismen eingegangen, die für das Schwarmverhalten sorgen. Im dritten Teil wird schließlich die Anpassung des Modells auf verschiedene Schwarmarten erklärt.

## 2. Grundlagen

Die Idee, die in diesem Modell steckt, ist zunächst die Aufteilung des Bewegungsverhaltens in mehrere Schichten. Eine mögliche Einteilung ist z.B. Planen, Steuern und Bewegung. In der Planung werden Grobziele vorgesehen, diese in der Steuerung dann in kleinere Abschnitte unterteilt und schließlich mit Hilfe der Bewegung realisiert.

Als Beispiel könnte man einen Fußgänger nehmen, der in einer Stadt von Punkt A nach Punkt B will. Er bestimmt dazu zunächst den ganzen Weg, der zurückgelegt

werden muss. Das ist in diesem Beispiel die Planung. Dass er nicht gegen andere Leute läuft, die richtigen Abzweigungen nimmt und so weiter, gehört zur Steuerung. Die Bewegung ist nun das tatsächliche Gehen, also einen Fuß vor den anderen setzen, nach links und rechts gehen und so weiter.

In diesem Modell wird vor allem auf die Steuerung und Teile der Bewegung eingegangen.

Um nun einen Schwarm zu simulieren benötigt man mehrere Objekte, die alle eine eigene Bewegung haben. Dies ist ähnlich dem, was ein Partikelsystem macht [3]. Partikelsysteme werden oft verwendet um dynamische Objekte zu simulieren, wie etwa Feuer oder Rauch. Man kann sagen, dass ein hier simulierter Schwarm in einer gewissen Weise die Erweiterung eines Partikelsystems ist, allerdings haben die Elemente des Schwarms keinen Lebenszyklus, wie die Partikel in einem Partikelsystem. Ein weiterer Unterschied ist, dass sich Tiere in Schwärmen komplexer Verhalten als in Punkte in Partikelsystemen und daher auch eine stärkere Interaktion innerhalb des Schwarms erfordern.

### 3. Grundlegende Verhaltensweisen der Bewegung

Um die Bewegung von Schwärmen zu simulieren bietet es sich an, zunächst die Bewegungen einzelner Schwarmmitglieder nachzubilden. Der erste Schritt hierbei ist nun ein „Fahrzeug“ zu entwerfen, welches dann im Folgenden die Verhaltensweisen verwenden soll. Dieses „Fahrzeug“ ist nicht festgelegt und kann beliebige Dinge darstellen, etwa einen Vogel, einen Fisch oder auch einen Toaster. Man benutzt im Zusammenhang mit Schwärmen für dieses abstrakte „Fahrzeug“ gerne den Begriff *Boid*, der von dem Begriff *bird-oid* kommt, also vogelähnliches Objekt.

#### 3.1 Ein Fahrzeugmodell

Ein sehr einfaches *Boid*-modell basiert auf einem punktförmigen Objekt. Diese Vereinfachung dient der Vereinfachung und erhält die Möglichkeit weiterhin beliebige Objekte einzusetzen, wobei dann dieses Modell variiert werden muss. Um nun mit diesem punktförmigen Objekt Bewegungen zu machen, müssen noch weitere Eigenschaften hinzugefügt werden. Nötig sind die aktuellen Informationen über den *Boid*, nämlich der Geschwindigkeitsbetrag, die Ausrichtung und die Position im Raum, sowie die festen Größen maximale Geschwindigkeit und maximale Beschleunigung. Der Geschwindigkeitsbetrag und die Ausrichtung werden zusammengefasst zu einem Vektor, den man als Geschwindigkeit bezeichnet. Ähnlich ist die Beschleunigung ebenfalls ein Vektor und beinhaltet ebenfalls die Ausrichtung. Nun kann man die für die Bewegung wichtigen Werte berechnen. Da in diesem Modell mit diskreten Zeiten gearbeitet wird, nämlich *Frames*, gestaltet sich die Berechnung der Werte relativ einfach. Diese Berechnung wird in jedem Frame der Simulation neu ausgewertet. Zunächst braucht man die aktuelle Beschleunigung, die durch die maximale Beschleunigung beschränkt wird. Die aktuelle Geschwindigkeit ergibt sich aus der alten Geschwindigkeit plus der Beschleunigung und ist beschränkt durch die maximale Geschwindigkeit. Die neue Position ist schließlich die alte Position plus die Geschwindigkeit. Dies funktioniert, da man die Geschwindigkeit relativ zur Dauer der *Frames* berechnet.

### 3.2 Verschiedene Verhaltensweisen

Im Folgenden werden einige grundlegende Verhaltensweisen erläutert, die es ermöglichen, dass sich ein *Boid* autonom bewegen kann. Danach werden noch weitere Verhaltensweisen, die sich auf die Interaktion von mehreren *Boids* beziehen, vorgestellt.

#### 3.2.1 Zielflug und Fliehen (*Seek and Flee*)

Realisiert wird das Verhalten *Zielflug* durch einen Zielvektor, der in Richtung des Suchziels zeigt. (*desired velocity* in der Abbildung 1). Die Länge dieses Zielvektors bestimmt, wie schnell die Richtung geändert wird.

Die Beschleunigung wird dann über aktuelle Geschwindigkeit minus Zielvektor berechnet. (*seek steering* in der Abbildung 1) und wird durch die maximale Beschleunigung beschränkt. Die neue Geschwindigkeit ist wieder alte Geschwindigkeit plus Beschleunigung, wie oben erläutert.

Das *Fliehen* Verhalten wird genauso berechnet, nur das der Zielvektor eine genau entgegengesetzte Richtung hat.

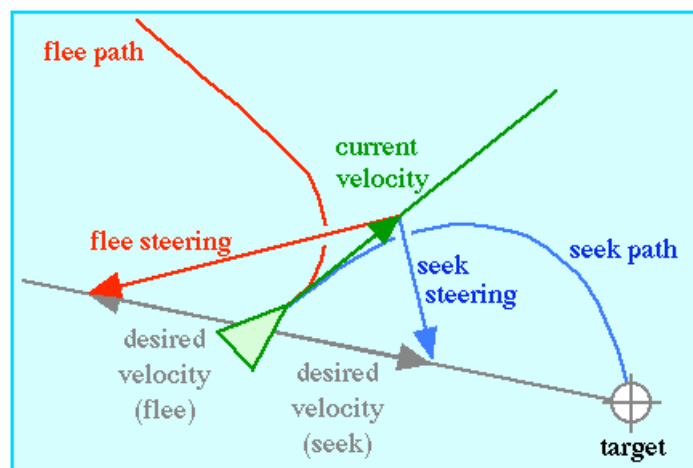


Abbildung 1: Suchen und Fliehen

Quelle: [2]

#### 3.2.2 Verfolgen und Vermeiden (*Pursuit and Evasion*)

Diese Verhalten sind *Zielflug* und *Fliehen* sehr ähnlich, nur das es sich hierbei um bewegende Zielpunkte handelt. Bei *Verfolgen* wird die Position des Ziels vorberechnet. Hierzu wird z.B. die Geschwindigkeit des Ziels auf dessen aktuelle Position addiert. Auf die neue Position wird nun einfach das *Zielflug* Verhalten angewendet. In jedem Frame wird die Berechnung mit der aktuellen Zielposition wieder neu evaluiert.

Wie auch bei *Zielflug* und *Fliehen* ist *Vermeiden* dasselbe Verhalten wie *Verfolgen*, nur das auf die neu berechnete Zielposition das *Fliehen* Verhalten angewendet wird.

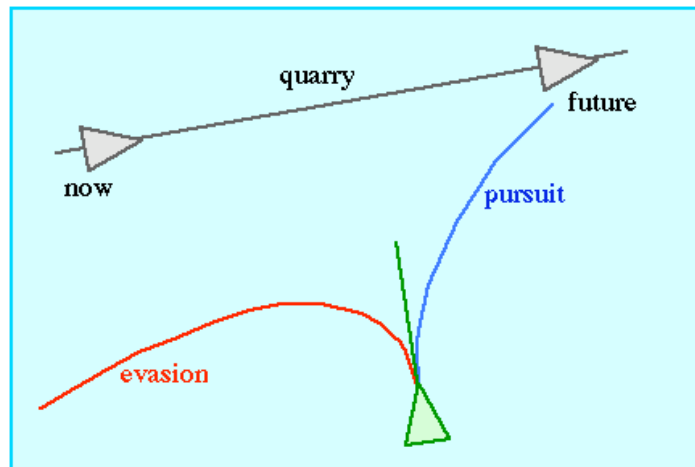


Abbildung 2: Verfolgen und Vermeiden  
Quelle: [2]

### 3.2.3 Paralleles Ausweichen (*Offset Pursuit*)

Eine Abwandlung des *Verfolgen* Verhaltens ist das so genannte *Paralleles Ausweichen*. Hier soll ein Objekt verfolgt werden, aber man möchte nicht direkt das Objekt ansteuern, sondern sich in einem gewissen Radius vorbeibewegen. Dazu wird bei der Berechnung des Zielpunktes wie bei *Verfolgen* dieser noch um den gewünschten Radius verschoben.

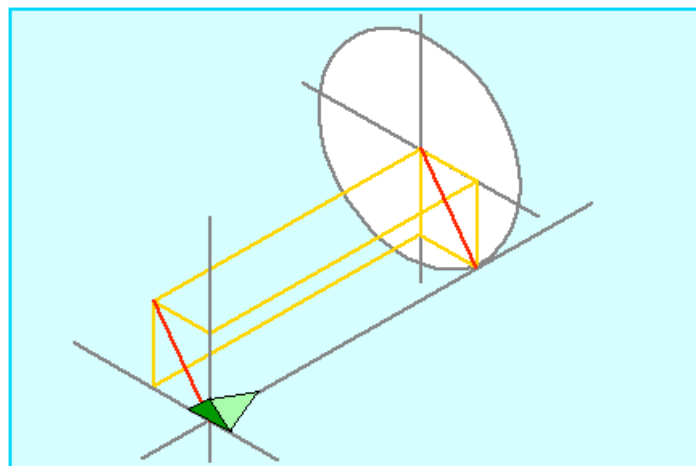


Abbildung 3: Paralleles Ausweichen  
Quelle: [2]

### 3.2.3 Ankommen (*Arrival*)

Ankommen ist wiederum eine Verhaltensweise, die auf *Zielflug* zurückgreift. Hier ist der Unterschied, dass die Bewegung langsam zu einem Stopp kommen soll während der *Boid* sich dem Ziel nähert.

Dazu verwendet man für den Zielpunkt einen „Abbrems“ Radius. Wenn man innerhalb des Radius ist, wird die maximale Geschwindigkeit z.B. bestimmt durch die maximale Geschwindigkeit des *Boids* multipliziert mit Entfernung zum Ziel  $d$ , geteilt durch den Radius  $r$ .

$$\tilde{V}_{\max} = \frac{V_{\max} * d}{r}$$

So wird die Bewegung immer weiter abgebremst und kommt am Ziel zu einem Stillstand.

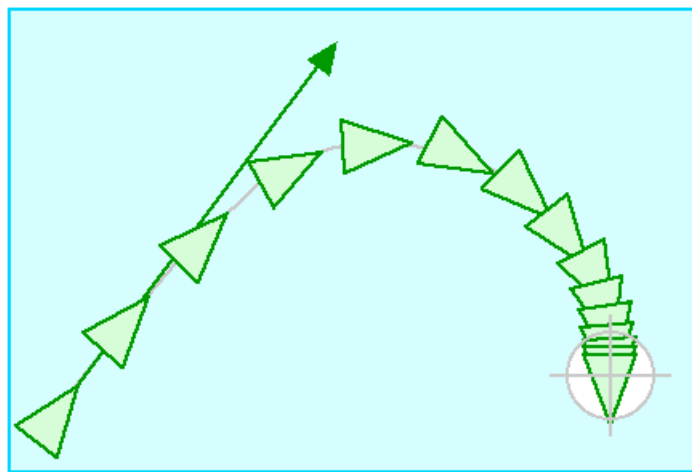


Abbildung 4: Ankommen  
Quelle: [2]

### 3.2.4 Ausweichen (*Obstacle Avoidance*)

Eine der wichtigsten Verhaltensweisen ist das Ausweichen. Als Vereinfachung wird angenommen, dass man alle Objekte als Kreise oder Kugeln beschreiben kann. Für die Realisierung dieses Verhaltens wird versucht immer einen freien Korridor vor dem *Boid* zu haben. Die Länge des freien Korridors hängt direkt von der Geschwindigkeit des *Boids* ab. Man prüft, ob ein Gegenstand ein Hindernis darstellt, indem man untersucht, ob eine Kugel mit einem bestimmten Radius um das Objekt diesen Korridor schneidet. Von den nun noch übrigen Hindernissen wird das als Gefährlichstes angesehen, das den Korridor als erstes vom *Boid* aus gesehen schneidet.

In Abbildung 5 ist zum Beispiel Objekt A gänzlich uninteressant, Objekt B und C sind Hindernisse und Objekt B wird als am gefährlichsten angesehen und daher wird der *Boid* im nächsten Frame nach links steuern.

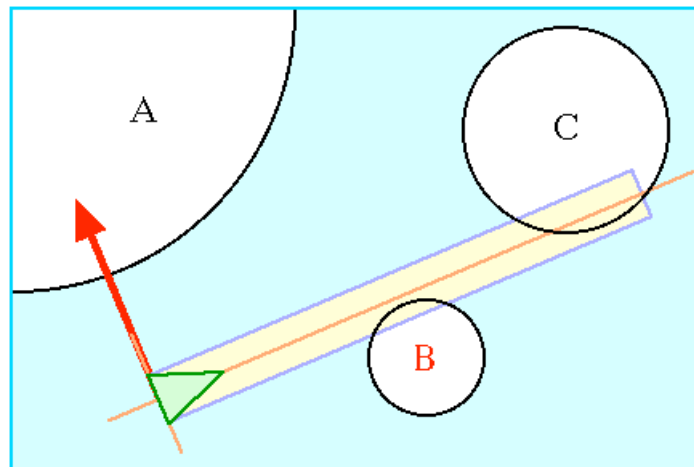


Abbildung 5: Ausweichen  
Quelle: [2]

### 3.2.5 Wandern (*Wander*)

Alle bisherigen Verhalten haben einen Zielpunkt oder ein Zielobjekt benötigt. Um nun noch eine Bewegung ohne direkt angegebenes Ziel zu ermöglichen bedient man sich des *Wanderns*. Man könnte nun einfach in jedem Schritt einen zufälligen Steuerungsvektor verwenden, wobei die Bewegung dann nicht sehr natürlich aussieht. Ein besseres Ergebnis erhält man, wenn man den Zielpunkt auf einen Kreis setzt. Dann wird in jedem Schritt ein Zufallswert berechnet und der Zielpunkt nach diesem Wert auf dem Kreis verschoben. Dadurch werden die Richtungsänderungen gemäßiger, als wenn direkt Zufallsvektoren als neue Geschwindigkeit verwendet werden, und es entsteht ein natürlicherer Eindruck der Bewegung.

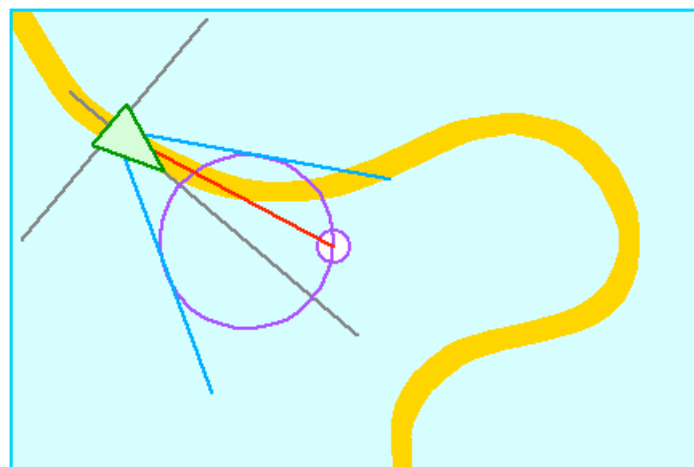


Abbildung 6: Wandern  
Quelle: [2]

### 3.2.6 Anziehung (*Cohesion*)

Bei *Anziehung* geht es darum einen gewissen Zusammenhalt zu erreichen. Das heißt, dass die *Boids* hier lieber zusammenbleiben. Dies wird realisiert indem man zunächst alle anderen Gruppenmitglieder findet die sich in der *Umgebung* befinden. *Umgebung* kann zum Beispiel definiert werden als ein Kreis um das Boid, der durch den Sichtwinkel eingeschränkt wird (Abbildung 8). Von allen so gefundenen Gruppenmitgliedern wird nun der Mittelpunkt berechnet und dieser Punkt dann wieder als Ziel für *Zielflug* verwendet.

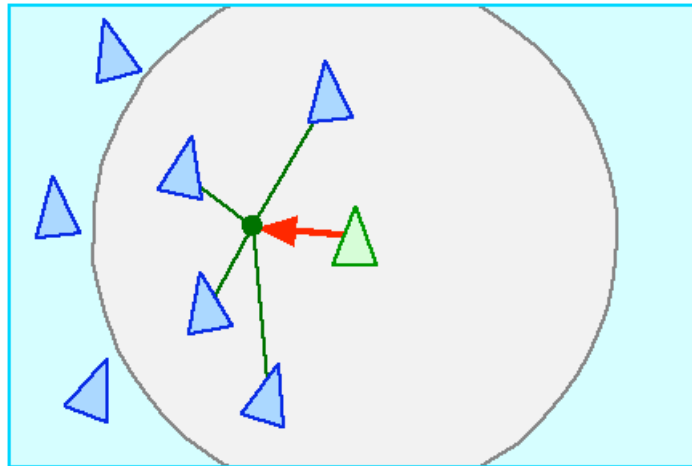


Abbildung 7: Anziehung  
Quelle: [2]

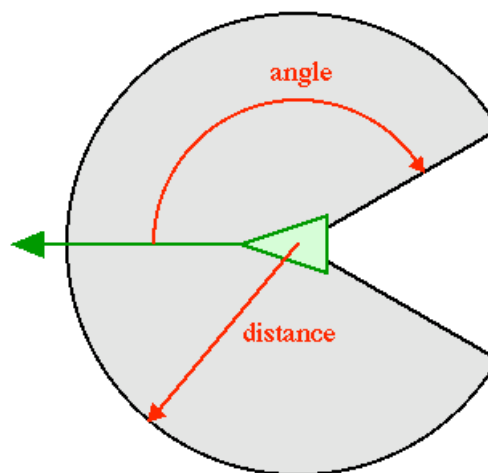


Abbildung 8: Umgebung  
Quelle: [2]

### 3.2.7 Abgrenzung (*Separation*)

Ein ähnliches Verhalten wie *Anziehung* ist *Abgrenzung*. Hier werden alle Vektoren von den Gruppenmitgliedern in der *Umgebung* des Boids addiert und

ergeben so den Zielvektor. So wird erreicht, dass die *Boids* im Schwarm nicht kollidieren, da immer ein gewisser Abstand gewahrt wird.

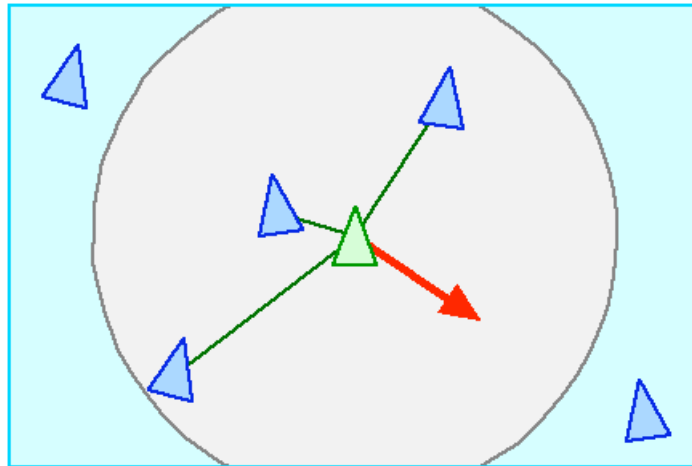


Abbildung 9: Abgrenzung

Quelle: [2]

### 3.2.8 Ausrichten (*Alignment*)

Auch hier spielt die *Umgebung* wieder eine wichtige Rolle. Hier wird der Geschwindigkeitsvektor aller in der *Umgebung* befindlicher Objekte gemittelt und die Differenz dieses Vektors mit dem aktuellen Geschwindigkeitsvektor ergibt dann den neuen Zielvektor. So gleichen sich sowohl Ausrichtung als auch Geschwindigkeit in der Gruppe immer mehr an.

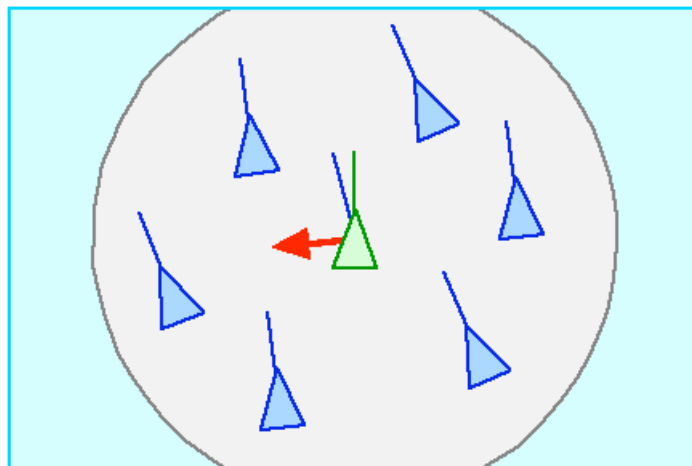


Abbildung 10: Ausrichten

Quelle: [2]

## 4. Schwarmverhalten

### 4.1 Einzelverhalten

Um einen Schwarm zu bilden müssen die Mitglieder im Großen und Ganzen drei Verhaltensweisen haben. Erstens müssen sie vermeiden mit anderen *Boids* zusammenzustößen. Außerdem sollten sie versuchen die Ausrichtung und die Geschwindigkeit aneinander anzugleichen und schließlich gibt es noch das Verhalten, nahe am „Zentrum“ des Schwarms zu sein.

Das Vermeiden von Kollisionen und Abgrenzung, zu einem gewissen Teil auch Ausrichtung, ermöglichen, dass sich die *Boids* frei im Schwarm bewegen können ohne zusammenzustößen. Abgrenzung verhindert, dass *Boids* gegeneinander stoßen und Ausweichen verhindert, dass der Schwarm gegen Hindernisse stößt. Daher ergänzen sich diese beiden Verhalten und erreichen einen in sich kollisionsfreien Schwarm. Die Realisierung der beiden Verhalten kann beispielsweise mithilfe der oben genannten Verhalten *Abgrenzung* und *Ausweichen* geschehen. Insbesondere zu Ausweichen und der Verwendbarkeit im Schwarmmodell siehe 4.4.

Für das dritte Bedürfnis ist der Begriff des Zentrums des Schwarms interessant. Im Normalfall sieht ein *Boid* nicht alle Anderen auf einmal, sondern nur die, die sich in der Nachbarschaft (siehe oben) aufhalten. *Boids* im Innern des Schwarms haben auf allen Seiten ungefähr gleich viele Nachbarn, daher ist die benötigte Beschleunigung zum Zentrieren gering und das Verhalten tritt in den Hintergrund. Am Rand des Schwarms befinden sich mehr Nachbarn auf der dem Schwarmzentrum zugewandten Seite und gibt es eine größere Beschleunigung in Richtung des Schwarmzentrums.

Dieser Ansatz einen Schwarm zusammenzuhalten ermöglicht auch, dass sich ein Schwarm aufteilt um ein Hindernis zu umgehen, wie es auch in natürlichen Schwärmen auftritt. In diesem Modell sieht jedes Mitglied nur seine eigene Nachbarschaft und „bemerkt“ oft gar nicht, dass es eine Aufteilung gibt. Bei einem einfachen Modell in dem z.B. nur alle *Boids* einem einzigen Mitglied folgen, kann so eine Aufteilung nicht auftreten, da die *Boids*, die auf der „falschen“ Seite des Hindernisses sind, zwar noch ein Schwarmverhalten zeigen, aber kein Ziel mehr haben.

### 4.2 Kombinieren von Verhalten

Das Hauptproblem in der Realisierung dieses Modells ist zunächst die Kombination der Verhalten. Berechnet man einfach das Mittel der drei Vektoren bekommt man dann ein Problem wenn man direkt vor einem Hindernis ist. In diesem Fall wäre der Vektor für das Ausweichen am wichtigsten. Dieser kann aber durch die anderen Vektoren gemindert werden und es wird dann wohl eine Kollision auftreten, was so ziemlich das ungünstigste ist was einem Tier in einem Schwarm passieren kann (abgesehen von gefressen werden...)

Der Ansatz in diesem Modell ist eine nach Prioritäten geordnete Liste von Steuervektoren. Die Prioritäten können dynamisch der Situation angepasst werden. Der Vektor für Ausweichen könnte also umso höhere Priorität erhalten, je näher ein Hindernis ist, und bei weit entfernten oder gar keinen Hindernissen wieder in der Priorität abgesenkt werden.

Die Vektoren werden nun nach ihrer Priorität geordnet addiert, wobei die Summe der Vektoren die maximale Beschleunigung nicht überschreiten darf. Diese Strategie funktioniert nun auch in extremen Situationen, da in dem Fall einer drohenden Kollision die gesamte Beschleunigung für den Vektor mit der höchsten Priorität verwendet wird und die anderen Bedürfnisse vernachlässigt werden können.

### 4.3 Wahrnehmung

Wie oben bereits erläutert basiert die Wahrnehmung der *Boids* hauptsächlich auf der Nachbarschaft, die ja durch einen Kreis bzw. eine Kugel beschrieben wird. Dies entspricht der Wahrnehmung von natürlichen Schwärmen eher als wenn alle *Boids* sämtliche Positionen und Geschwindigkeiten aller anderen Objekte „wissen“ würden. Als noch bessere Annäherung werden nähere *Boids* stärker berücksichtigt als die, die weiter entfernt sind. Dadurch erhält man eine verhältnismäßig gute Annäherung an die tatsächliche Wahrnehmung eines Tieres. Weitere Verbesserungen an dem Wahrnehmungsmodell wären die Nachbildung des tatsächlichen Gesichtsfeldes (mit stereoskopischem Sehen im überlappenden Bereich usw.) und natürlich die Simulation des Sehens [4]. Das Simulieren von Sehen ist aber viel zu komplex und nicht unbedingt nötig für die Simulation eines Schwarmverhaltens.

### 4.4 Ausweichen im Schwarm

Das Ausweichen im Schwarm bedarf keiner besonderen Implementierung, sondern wird dadurch realisiert, dass die einzelnen *Boids* das in 3.2.4 beschriebene Verhalten verwenden. Dieses Verhalten funktioniert zum Ausweichen gut, allerdings nur für Objekte die durch Kugeln angenähert werden können. In einer Schwarmsimulation sollte man genauer werden und muss daher dieses Verhalten verfeinern. Es sollten andere geometrische Objekte verwendet werden und damit stellt sich zunächst einmal das Problem, in welche Richtung man Steuern muss, um nicht mit dem Hindernis zu kollidieren. Dieses Problem ist bei Kugeln sehr einfach zu lösen, da man nur die Richtung zum Zentrum der Kugel finden muss. Für regelmäßige Formen funktioniert ein anderer Ansatz, der ähnlich einfach ist.

Um einem einfachen geometrischen Körper auszuweichen wird die nächstgelegene Kante gesucht und der eigenen Steuervektor in diese Richtung anpasst, wie in Abbildung 11. Dieses Problem gewinnt allerdings an Komplexität, wenn man beliebige, nichtregelmäßige Objekte betrachtet. Eine allgemeine Implementierung für solche Objekte ist wieder ein komplexeres Problem, weswegen man in diesem Modell eine Approximation auf einfache, geometrische Formen verwendet.

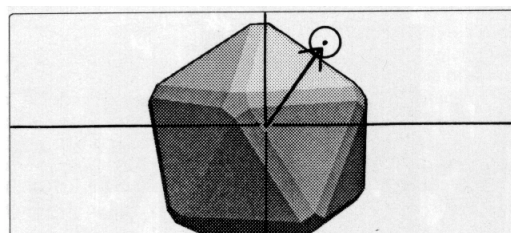


Abbildung 11: Ausweichen an geometrischen Objekten

Quelle: [1]

## 4.5 Gesteuerte Schwärme

Wenn man dieses Schwarmmodell in der Animationstechnik verwendet möchte man in den meisten Fällen eine gerichtete Bewegung oder zumindest einen Zielpunkt für den Schwarm festlegen.

Dies wird in dem Modell über globale Variablen realisiert. Entweder als allgemeine Richtungsangabe (z.B. „Vögel fliegen im Winter in den Süden“) oder als direkten Zielvektor des Schwarms. Diesen direkten Zielvektor kann man natürlich von Frame zu Frame anpassen und so quasi eine Route durch die Simulation beschreiben.

Auch eine Möglichkeit ist, nicht allen *Boids* einen Zielpunkt mitzugeben, sondern nur einer Gruppe von *Boids*. Der Extremfall, nämlich, dass die „Führungsgruppe“ nur aus einem *Boid* besteht, ist allerdings nicht sehr praktikabel und führt vor Allem bei Spaltungen des Schwarms zu unnatürlichem Verhalten.

Eine weitere Möglichkeit ist, ein Vektorfeld über das gesamte Simulationsgebiet zu legen. Die *Boids* können dann den Vektoren dieses Felds folgen und so durch die Simulation geleitet werden.

## 5. Spezielle Anpassung des Modells

Das dargestellte Modell eignet sich für die Simulation aller Objekte, die sich in größeren Gruppen bewegen. Im Allgemeinen sind das Vögel, Fische und Landlebewesen. Für Landlebewesen werden sämtliche Vektoren nur im zweidimensionalen berechnet während sich Fische und Vögel dreidimensional bewegen. Der größte Unterschied zwischen Fischen und Vögeln liegt in Faktoren wie der angestrebten Entfernung zu den anderen *Boids*, der Maximalbeschleunigung und der Maximalgeschwindigkeit.

Natürlich sind dabei auch weitere Anpassungen im Bewegungsverhalten nötig. So ist bei fliegenden Objekten eine Kurve immer mit einer Seitwärtsneigung verbunden ähnlich dem „In-die-Kurve-legen“ von Motorradfahrern. Bei fliegenden Tieren nennt man das „Banking“.

Außerdem gibt es einige Anwendungen, die sich nicht auf den ersten Blick erschließen. Ein Beispiel wäre der Verkehrsfluss auf Strassen. Auch wenn hier die Fahrzeuge fest in den Fahrbahnen bleiben (zumindest sollten sie das), ist das Verhalten, das Kollisionen vermeidet, dem hier vorgestellten sehr ähnlich.

Andererseits ermöglicht ein solches Modell auch große Menschenmengen in Filmen relativ natürlich nachzubilden ohne die entsprechende Anzahl von Schauspielern zu beschäftigen.

## 6. Zusammenfassung

In dieser Arbeit wurde ein Modell vorgestellt, das aufbauend auf sehr einfachen Grundverhalten ein eindrucksvoll natürliches Schwarmverhalten generiert. Dazu wurden zunächst die Verhaltensweisen dargelegt und dann deren Anwendung im tatsächlichen Modell. Dieses Modell ist sehr allgemein gehalten, um möglichst viele Anpassungen zu ermöglichen. Einige Anpassungsmöglichkeiten wurden im letzten Teil noch näher erläutert.

Im Allgemeinen ist das hier vorgestellte Modell sehr gut, vor allem weil die eingesetzten Methoden und Verhalten sehr einfach sind und erst durch das Zusammenwirken von mehreren Verhalten entsteht das komplexe, natürliche Verhalten eines Schwarms. Ein weiterer positiver Punkt sind die Ausbaumöglichkeiten des Modells, da hier nur ein allgemeines Grundgerüst geliefert wird, das mit einfachen Anpassungen auf viele Anwendungen passt. Auch finden Teile des Modells (insbesondere die verschiedenen Verhaltensweisen und deren Implementierung) auch außerhalb des Modells eine Anwendung. Zum Beispiel wird die natürliche Simulation des Verkehrs einer Stadt in den Computerspielen *Midtown Madness 2* und *Midnight Club* auf Basis solcher Verhaltensweisen gestaltet [5].

Es ist in jedem Fall faszinierend einen auf diesem Modell generierten Schwarm zu beobachten, da die „chaotische Dynamik“ eines Schwarms sehr gut simuliert wird.

## 7. Literaturliste

- [1] Craig W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", published in *Computer Graphics*, 21(4), July 1987, pp. 25-34. (ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California, July 1987.)  
<http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87>
- [2] Craig W. Reynolds, "Steering Behaviors For Autonomous Characters",  
<http://www.red3d.com/cwr/steer/gdc99/>
- [3] Reeves, W., T., "Particle Systems-A Technique for Modeling a Class of Fuzzy Objects," *acm Transactions on Graphics*, V2 #2, April 1983. and reprinted in *Computer Graphics*. V17 #3, July 1983, (acm SIGGRAPH '83 Proceedings), pp. 359-376.
- [4] Pinker, S. (editor), *Visual Cognition*, The MIT Press, Cambridge, Massachusetts, 1985
- [5] [http://www.gamasutra.com/features/20010124/adzima\\_01.htm](http://www.gamasutra.com/features/20010124/adzima_01.htm)